A Diner's Guide to Evaluating a Framework for Ubiquitous Computing Applications

Mary Theofanos *Jean Scholtz*

National Institute of Standards and Technology
100 Bureau Drive, Gaithersburg, MD. 20899
mary.theofanos@nist.gov
jean.scholtz@nist.gov

Abstract

There is a clear need for evaluation methodologies specifically suited to ubiquitous computing applications. Here we investigate a user evaluation framework we proposed earlier which draws upon traditional desktop methods, but carefully adapts them based on our experiences with ubiquitous architectures. We test and clarify the criteria in our methodology by examining the utility and applicability of the framework to an existing commercial ubiquitous application for restaurant ordering at the tableside. We analyzed its functionality by discussing design principles with its software developers, and interviewed wait staff as well as restaurant managers to understand its impacts on the workflow and business processes. We conclude that the proposed framework does contain appropriate metrics to assess whether good design principles were achieved and if the designed system will produce the desired user experience.

1 Introduction

Ubiquitous computing applications are diverse in nature and are very challenging to evaluate. These systems are based on devices where speech, gesture, and even physical interactions, in addition to the traditional graphical and text-based user interfaces, can be used as interaction modalities. Examples of these range from personal digital assistants and cell phones, large touch screen displays, and in-vehicle devices, to smart laboratories and spaces. Further, these interactions are made even more complex because many of the applications involve access to large amounts of diverse information, collaboration among users, and multi-tasking in distracting environments. Ubiquitous computing applications are by their very definition embedded in the real world. Users of these computing applications are engaged in other tasks and need the application to contribute to the overall experience. Critical to the success and acceptance of these ubiquitous systems is the human computer interaction (HCI). However, it is difficult to determine what makes for a good design and a successful interaction because evaluation methodologies and metrics are in their infancy for these new types of systems. The creation of a common approach to evaluation will require a good deal of experimentation and measurement by the research community. This can be facilitated by the establishment of a consistent terminology, an initial set of metrics, and a framework in which researchers can share and learn from each other's evaluations. The benefits from such an approach will be validated metrics, effective discount evaluation techniques, and design guidelines which can then be put to use to improve the human computer interaction of these systems.

2 Background

Most traditional desktop computing applications assume one user per application at any given time. Moreover, the typical user environment assumes a user seated at a desk with a monitor, a keyboard, and mouse. Thus traditional usability evaluations have focused on three metrics: efficiency, effectiveness, and user satisfaction (ISO 92411-11). Efficiency measures the resources expended in relation to the accuracy and completeness with which users achieve goals. Effectiveness measures the accuracy and completeness with which users achieve specified goals. Satisfaction measures freedom from discomfort, and positive attitudes towards the use of the product.

As a result a number of user evaluation techniques have been developed for desktop computing applications including heuristic evaluations (Nielsen, 1993), user testing (Dumas & Redish, 1994), cognitive walkthroughs (Lewis, Polson, Wharton & Rieman, 1990), and modeling (Card, Moran, & Newell, 1980). These evaluation techniques have resulted in the development of guidelines (Koyani, Bailey & Nall, 2003) for desktop computing applications that have had a very positive effect on the implementation of more usable systems. Guidelines allow developers to take the guesswork out of initial interface design.

These guidelines, techniques and metrics have proven very helpful in evaluating traditional desktop computing applications, but they are not sufficient for ubiquitous applications that place more of an emphasis on values, emotion, privacy, trust and other social aspects of computing.

3 Framework for User Evaluations of Ubiquitous Computing

An initial framework for HCI evaluations of ubiquitous computing applications has been proposed (Scholtz & Consolvo, 2004). This framework identifies a set of user evaluation areas with associated metrics and measures. Measures are defined as observable values. Associating meaning to those values by applying human judgement results in metrics. Table 1 identifies the 9 user evaluation areas (UEAS), the associated metrics and measures.

Table 1: User Evaluation Areas (UEAs) for Ubiquitous Computing Applications

UEA	Metric	Conceptual Measures
Attention	Focus	Number of times a user needs to change focus due to technology; number of different displays/actions a user needs to accomplish, or to check progress, of an interaction; number of events not noticed in an acceptable time
	Overhead	Percent of time a user spends switching foci; workload imposed on the user due to changing focus
Adoption	Rate	New users/unit of time; adoption rationale; technology usage statistics;
	Value	Change in productivity; perceived cost/benefit; continuity for user; amount of user sacrifice
	Cost	User willingness to purchase technology; typical time spent setting up and maintaining the technology
	Availability	Number of actual users from each target user group; technology

		supply source; categories of users in post-deployment
	Flexibility	Number of tasks user can accomplish that are not originally envisioned; user ability to modify as improvements and features are added
Trust	Privacy	Type of information user has to divulge to obtain value from application; availability of the user's information to other users of the system or third party
	Awareness	Ease of coordination with others in multi-users application; number of collisions with activities of others; user understanding about how recorded data is used; user understanding inferences that can be drawn about him or her by the application
	Control	Ability of users to manage how and by whom their data is used; types of recourse available to user in the event that the data is misused
Conceptual Models	Predictability of application behaviour	Degree of match between user model and behaviour of application
	Awareness of application capabilities	Degree of match between user's model and actual functionality of the application; degree of match between user's understanding of his or her responsibilities, system responsibilities, and the actual situation; degree to which user understands the application's boundary
	Vocabulary awareness	Degree of match between user's model and the syntax used by the application
Interaction	Effectiveness	Percentage of task completion
	Efficiency	Time to complete a task
	User Satisfaction	User rating of performing the task
	Distraction	Time taken from the primary task; degradation of performance of primary task; level of user frustration
	Interaction transparency	Effectiveness comparisons on different sets of I/O devices
	Scalability	Effectiveness of interactions with large numbers of users
	Collaborative interaction	Number of conflicts; percentage of conflicts resolved by the application; user feelings about conflicts and how they are resolved; user ability to recover from conflicts
Invisibility	Intelligibility	User's understanding of the system explanation
	Control	Effectiveness of interaction provided for user control of system initiative
	Accuracy	Match between the system's contextual model and the actual situation; appropriateness of action; match between the system action and the action the user would have requested
	Customization	Time to explicitly enter personalization information; time for the system to learn and adapt to the user's preferences
Impact and Side Effects	Utility	Changes in productivity or performance; changes in output quality
	Behaviour changes	Type, frequency, and duration; willingness to modify behaviour or tasks to use application; comfort ratings of wearable system components

	Social acceptance	Requirements placed on user outside of social norms; aesthetic ratings of system components
	Environment change	Type, frequency, and duration; user's willingness to modify his or her environment to accommodate system
Appeal	Fun	Enjoyment level when using the application; level of anticipation prior to using the application; sense of loss when the application is unavailable
	Aesthetics	Ratings of application look and feel
	Status	Pride in using and owning the application; peer pressure felt to use or own the application
Application Robustness	Robustness	Percentage of transient faults that were invisible to user
	Performance speed	Measures of time from user interaction to feedback for user
	Volatility	Measures of interruptions based on dynamic set of users, hardware, or software

Traditional usability evaluations focus on users, but this framework also emphasizes stakeholders. As defined by Friedman et al. (Friedman, Kahn, & Borning, 2001) direct stakeholders interact with the application and/or its output in a direct way, while indirect stakeholders are affected by the application in some meaningful way although not directly. Consider a cell phone, the direct stakeholder (DS) is the person who uses the cell phone and makes and receives calls from it. The indirect stakeholders of the cell phone include people who receive calls from the DS, people who call the DS, people with the DS when using the cell phone, people around the DS but not with the DS. Thus to use the framework, evaluators must identify the direct stakeholders of the ubiquitous application.

This initial framework must now be applied to next generation interactive applications and its utility evaluated and validated. The following questions can be addressed by case studies:

- Is the framework complete? Are there any user evaluation areas that are missing?
- Is it useful?
- Does it provide metrics and measures that can differentiate systems?
- Can the framework be used to predict which systems will be useful and accepted by users?
- The current metrics primarily focus on direct stakeholders -- should we consider including additional metrics for indirect stakeholders?
- What are the interactions/correlations between the different user evaluation areas? Which evaluation areas are appropriate for which categories of ubiquitous computing applications? Are different evaluation areas applicable or have more weight depending on the category of ubiquitous computing applications? For instance, Salvador et al, (Salvador, Barile & Sherry, 2004) and Eriksen (Eriksen, 2002) found that the invisibility of certain ubicomp applications deters adoption and acceptance and therefore it needs to be countered with visibility and accountability.

4 Case Study

Our objective is to look at a number of diverse ubiquitous computing applications to determine if the proposed framework contains the appropriate evaluation areas. Given that, the next step is to assess utility. What might other ubicomp developers learn from an evaluation of similar applications? In the following section we describe a restaurant application we examined, look at the applicable sections of the framework, and draw some conclusions about the utility.

4.1 A Handheld Ordering System

In this case study we applied the framework to a handheld restaurant order entry system that relies on handwriting recognition. The application, developed by Action Systems Inc. and domain expert and Chief Executive Officer, Alex Malison (Stanford, 2003), essentially mimics the little green order pad that servers have traditionally used to interact with customers. [1] Thus the Write-On Handheld is a wireless system used by servers to process customer orders at tableside as shown in Figure 1. In this application the direct stakeholders are the servers; the indirect stakeholders include the customers, the kitchen and bar staff, and the manager and restaurant owner.

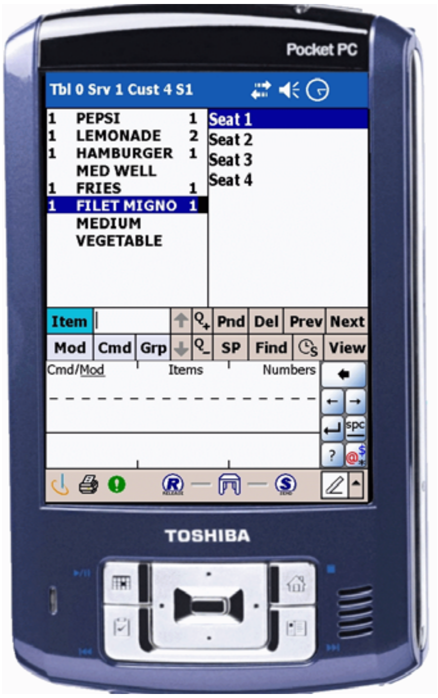


Figure 1: A Handheld Ordering System

4.1.1 Attention

The design team recognized the importance of not just moving the paper ordering pad to the handheld. However, they did want to retain some of these properties. Paper is flat; as a result the

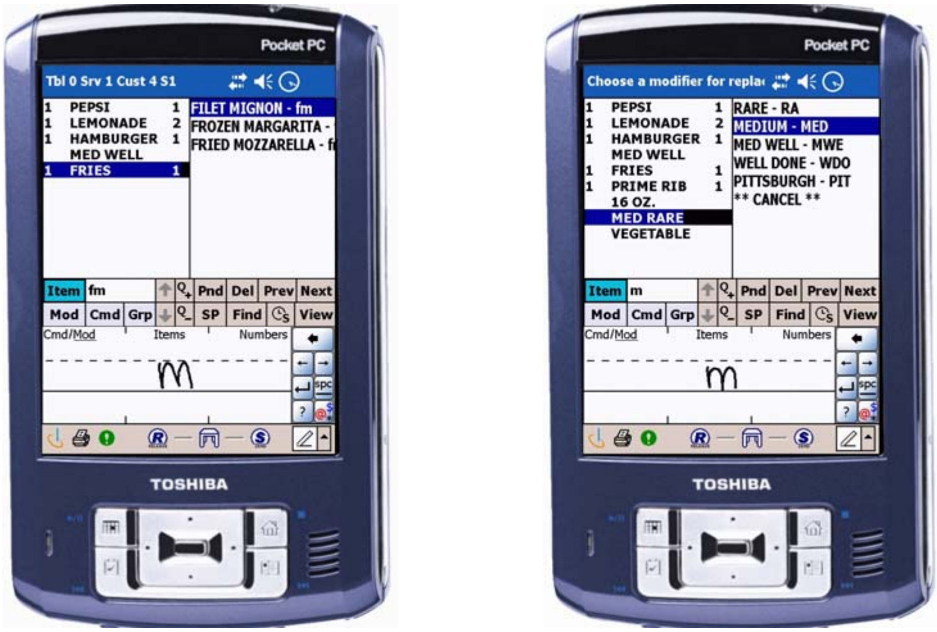
[1] Any commercial product identified in this document is for the purpose of describing a ubiquitous computing application to evaluate the framework. This identification does not imply any recommendation or endorsement by NIST.

handheld ordering interface should be flat. When using a traditional order pad a server doesn't flip back and forth between pages to take a customer's order.

The interface is segmented both for input and for display. The first takes advantage of the segmented pen entry fields on the handheld assigning the separate areas at the bottom of the handheld for capital letters, lowercase letters, and numeric data to differentiate commands. These assignments provide a clear context for servers who order items by printing in one area and listing quantities in another. The display area is also segmented. Ordered items are always displayed on the upper left of the screen while the upper right displays the lists of pending decisions such as preparation instructions and side orders. Thus, the order grows in the left quadrant, as decision lists cycle in the upper right quadrant. These design strategies provide a clear context for users who know exactly where to look and write on the screen at all times. The UEA of *attention* assesses this design principle. An evaluation of the time servers had to spend switching foci would have been appropriate here.

4.1.2 Interaction

The application was designed to reduce the number of menus, pen strokes and complexity. The design team developed an abbreviation scheme that relies on the first letter of each word to narrow the choices quickly. For example, entering "fm" results in a list that includes filet mignon as well as other items that begin with the "f" and "m" but now the list of items is small and fits easily on the handheld screen with minimal scrolling as shown in Figure 2. This makes ordering very efficient.



Tbl 0 Srv 1 Cust 4 S1

1 PEPSI 1
1 LEMONADE 2
1 HAMBURGER 1
MED WELL
1 FRIES 1

FILET MIGNON - fm
FROZEN MARGARITA -
FRIED MOZZARELLA - f

Item fm

Choose a modifier for repla

1 PEPSI 1
1 LEMONADE 2
1 HAMBURGER 1
MED WELL
1 FRIES 1
1 PRIME RIB 1
16 OZ.
MED RARE
VEGETABLE

RARE - RA
MEDIUM - MED
MED WELL - MWE
WELL DONE - WDO
PITTSBURGH - PIT
** CANCEL **

Item m

Figure 2: Segmented Interface for Display and Input

The design team used iterative development and beta restaurant sites to observe the application in action and collect data on *efficiency, effectiveness*, and *user satisfaction*.

Distraction is a critical factor for the servers – their primary task is to focus on and serve the customers and make them happy-- the technology cannot interfere. With the old order entry stations servers captured the orders on paper and then re-entered the orders away from the customer. Now they are interacting directly with the customer and the interface at the same time. Time is critical and the interface cannot cause anxiety in front of the customer. By observing the face and eye contact with the customer and the level of conversation observers can determine if the waiter can use the technology and still maintain the same level of “interaction" with the diners.

Scalability could also be a concern in large restaurants. How many waiters can be supported at once? In general, wireless access points support lots of people. The restaurant we visited had a number of access points due to the layout of the facility.

4.1.3 Impact and Side Effects

Utility – The restaurants that have introduced the new system have seen measurable improvements in productivity and performance as well as quality. According to one restaurateur in Maryland (Stanford, 2003), the handheld order system has:

- Reduced order errors from several meals per night to one every two nights
- Resulted in smaller payrolls
- Allowed servers to spend more time on the floor resulting in:
    - Opportunity to sell more drinks and desserts
    - Customers receive their meals faster
    - Increased tips
    - Ability to turn the tables faster

As a result he reported that sales of wine are up 12%, desserts 14%, and they sell more meals because the tables turn faster so overall revenue is up 15%.

Other factors that could be considered include *social acceptance* and *behavior changes,* both from the viewpoint of the waiters and the customers. Will people not want the job because they have to use the technology? Because it hangs on your belt, will restaurants change what the staff wears? From the viewpoint of the customers: do they like the staff using these? It has changed the structure of the server staff in some restaurants. Runners are now used to bring drinks to the table as the drinks are ready by the time (or before) the waiter finishes telling customers about the specials.

4.1.4 Adoption

The fixed touch screen menu system point of sale (POS) paradigm is embedded in the restaurant industry today. The status quo is powerful – especially in a high turnover industry where the underlying conceptual model is the same from touch screen system to touch screen system irrespective of the restaurant. The *cost* of training the staff is an issue. These systems all use drill down menus. The new interface had to be easy to learn to use. This led to the decision to eliminate the need to learn graffiti. Another driving force was to eliminate/minimize key strokes. The developers estimate it takes about one week to be comfortable using this in front of customers. However, it is also true that not all servers at an establishment have to use the handheld. A server who does not will be less productive but this could be a solution for new servers who do not yet feel comfortable using the device in front of customers.

The design team understood the market forces, the status quo and the inertia to change approaches would be a significant hurdle, knew the application had to be not only better than the existing POS systems but also better in every way than the little green ordering pad. So they had to improve every major task in the workflow, had to provide additional *flexibility* and *value*. There is no doubt on cost savings, that's not the issue, according to Malison (Malison, 2004), you have to get the restaurant management to accept the new interface and paradigm shift. So the impact and side effects may not be enough to influence adoption and overcome inertia.

Additional *costs* include the cost of initial setup. Someone must enter all the menu items. A downloadable bar menu is available with all sorts of drinks included. There is also the issue of maintaining the menus, specials must be included daily and wait staff must know the items and their abbreviations. This affects the servers as well. However, a search feature can be used to bring up; for example, all chicken dishes and provide the abbreviations to help locate new items.

The servers do need to be trained to use the handheld devices and need to practice before they feel comfortable using this in front of customers. Not all restaurants will have staff dedicated enough to do this, nor the resources to pay staff for training time.

4.1.5 Conceptual Model

The underlying conceptual model of a POS restaurant system is implemented in the restaurant industry today with the touch screen systems. Thus the *predictability of the applications behavior*, *awareness of the application capability,* and *vocabulary awareness* are not significantly different. However, the handheld user interface is based on a different conceptual model than the drill down menu screens of the touch pad systems. The design concepts of flattening the hierarchy of the interface and segmenting the interface were critical to acceptance of the new conceptual input model.

4.1.6 Application Robustness

Since the system is wireless coverage of the system should also be examined. What happens if diners sit outside? Will the transmitters reach? The POS terminal is always an option if the handheld is not working. Reboot time is extremely fast if something happens to one of the handhelds and orders are generally not stored on the device so little is lost. However, a new feature allows an order to be placed and setting a time for it to be sent. For example if you want to order a takeout – but want it ready after you have finished eating, then the order could stay on the handheld until the timer goes off. In this case, if the handheld reboots the timed order could be lost. To prevent losing an order, all data related to the order, including the timed order instructions are sent to the main server, where it is stored and acted upon at the specified time.

The hardware's robustness was also a concern and the design team mentioned a few characteristics of the hardware that they addressed. For instance battery life of the handheld– it was essential that it last throughout the shift. Effective backlighting of the screen was also required for evening shifts.

4.1.7 Appeal

In this application it is difficult to separate social acceptance from appeal. But the *aesthetics* should be considered and might be an issue here. Does the actual device look appealing? Does it fit in different types of establishments? The restaurant we visited was an upscale facility. The servers were formally dressed and the sleek look of the handheld fit in very well. To address theft and misplacement of the handheld, servers are encouraged to secure the handheld to the server's belt. This was no longer done at the location we visited and would have looked out of place in that establishment. The servers told us that the device was a conversation topic for many customers.

We discussed the time needed to learn to use the device effectively. The developers included a game on the handheld that helps train the staff giving a "fun" atmosphere

5 Conclusions

We concluded that the current framework was quite adequate for specifying the important evaluation areas for this ubicomp application. In this case the areas applicable are:

- Attention
- Interaction
- Adoption
- Conceptual models
- Impact and side effects
- Appeal
- Robustness

The framework does not specifically call out a UEA of training. This could certainly be assessed under cost in the adoption UEA as we did in this case study. However, it might be better to specifically include this in the framework as a UEA.

What about the utility of the framework? From the point of view of the developers of this application, the attention, interaction, impact and side effects covered their primary design rationale. The framework would have served to point out issues in other areas that they should be concerned with. For example, how does the handheld fit in different types of establishments?

What can developers of other ubicomp applications learn? If we conducted evaluations in these UEAs, we would have information about:

- Efficient design for small displays
- The number of side effects possible from a device inserted in an established process
- Considerations for making adoption lower cost, such as training aids and providing generic menu templates
- The importance of looking at the impact of the device on the servers, the restaurant management, and the customers

While a number of ubicomp applications need to be investigated to give us more input about the use of the proposed framework, we conclude that in this case, the framework was a good fit for the application. The UEAs could provide usability professionals with areas that should be considered. Developers of other ubicomp applications could also benefit from metrics in these UEAs.

References

Card, S.K., Moran, T.P., & Newell, A. (1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, 23(7), 396-410.

Dumas, J.S., & Redish, J.C. (1994). A practical guide to usability testing. New Jersey: Ablex Publishing.

Eriksen, E. (2002). Designing for accountability. *Proceedings of NordiCHI'02*, 177-186.

Friedman, B., Kahn, Jr., P.H., & Borning, A. (2001). *Value Sensitive Design: Theory and Methods*, tech. report 02-12-01, University of Washington, Dec. 2001.

Koyani, S., Bailey, R. & Nall, J. (2003). Research-based web design & usability guidelines. Retrieved August 25, 2004, from http://usability.gov/pdfs/guidelines_book.pdf

Lewis, C., Polson, P., Wharton, C., & Rieman, J. (1990). Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. *Proceedings of the Conference on Human Factors in Computing Systems,* 235-242.

Malison, A., & Bolling, D. (personal communication, July 23, 2004).

Nielsen, J. (1993). Heuristic evaluation. In J. Nielsen & R.I. Mack (Eds.), *Usability Inspection Methods. (pp*. 25-62). New York: John Wiley & Sons.

Salvador, T., Barile, S., & Sherry, J. (2004). Ubiquitous computing design principles: Supporting human-human and human-computer transactions. *Proceedings of the Conference on Human Factors in Computing Systems,* 1497-1500.

Scholtz, J., & Consolvo, S. (2004). Toward a framework for evaluating ubiquitous computing applications. *Pervasive Computing*, April-June, 82-88.

Stanford, V. (2003). Pervasive computing puts food on the table. *Pervasive Computing*, January – March, 15-20.

Restaurant Manager. Write on handheld. Retrieved September 10, 2004, from http://www.actionsystems.com/